Decoding Reed-Solomon codes up to the Sudan radius with the Euclidean algorithm

Alexander Zeh, Wenhui Li

Decoding Reed–Solomon Codes up to the Sudan Radius with the Euclidean Algorithm

Alexander Zeh and Wenhui Li
Department of Telecommunications and Applied Information Theory
University of Ulm, Germany
{alexander.zeh,wenhui.li}@uni-ulm.de

***Abstract*—We modify the Euclidean algorithm of Feng and Tzeng to decode Reed–Solomon (RS) codes up to the Sudan radius. The basic steps are the virtual extension to an Interleaved RS code and the reformulation of the multi–sequence shift–register problem of varying length to a multi–sequence problem of equal length. We prove the reformulation and analyze the complexity of our new decoding approach. Furthermore, the extended key equation, that describes the multi–sequence problem, is derived in an alternative polynomial way.**

***Index Terms*—Reed–Solomon (RS) codes, Interleaved Reed–Solomon (IRS) codes, Euclidean algorithm, Shift–Register synthesis**

I. Introduction

The Euclidean algorithm (EA) can be used to decode RS codes up to half the minimum distance. The proof was given by Sugiyama, Kasahara, Hirasawa and Namekawa (SKHN,[1]) in 1975 and the equivalence to the Berlekamp–Massey (BM, [2]) algorithm, synthesizing the shortest linear feedback shift–register (LFSR) generating one sequence $\mathbf{S}$, is widely accepted.

Feng and Tzeng (FT) generalized the SKHN approach in 1989 [3] and the BM algorithm in 1991 [4] to multiple sequences. Both generalizations synthesize the shortest linear feedback shift–register generating of s sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(s-1)}$ of equal length N.

While in [5] a modification of the 1991–Feng–Tzeng approach for the case of many sequences with different lengths was given, the adaption of the Euclidean–based algorithm (1989–FT) is still missing. Feng–Tzeng's Euclidean algorithm was deduced by Wang [6] through the extension of the concept of a Euclidean ring to arbitrary modules.

In our contribution we use FT's Euclidean algorithm to decode Interleaved Reed–Solomon (IRS) codes ([7], [8]), more specifically a virtual extension to an IRS code that has an equivalent error–correcting capability as Sudan's list–decoding algorithm [9], [10].

The concepts of IRS codes and FT's Euclidean algorithm are explained in the following section. For the latter one we use a new alternative derivation of the extended key equation. We recall the basic principle of the virtual extension to an IRS code and the corresponding decoding problem in Section III. The reformulation of the multi–sequence problem of varying length is shown and proved in Section IV. In Section V we analyze the complexity for our new approach fitted to the virtual extension and outline the reformulation for the more general case in Section VI. Section VII concludes our contribution.

II. Feng–Tzeng's Euclidean Algorithm for Homogeneous Reed–Solomon Codes

A. Basic Idea of Interleaved RS Codes

Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be the nonzero distinct elements (code–locators) of the finite field $\mathbb{F} = GF(q)$. $\mathcal{L} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is the set of the code–locators. Denote

$$f(\mathcal{L}) = (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$$

for a given univariate polynomial $f(x) \in \mathbb{F}[x]$.

A Reed–Solomon code $\mathcal{RS}(n,k)$ over a field $\mathbb{F}$ with $n < q$ is given by

$$\mathcal{RS}(n,k) = \{\mathbf{c} = f(\mathcal{L}) : f(x) \in \mathbb{F}_k[x]\}, \tag{1}$$

where $\mathbb{F}_k[x]$ stands for the set of all univariate polynomials with degree less than k and the indeterminate x.

RS codes are known to be maximum–distance separable (MDS), i.e., their minimum Hamming distance is $d = n-k+1$. Let $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ be the received word and $r(x) = \sum_{i=1}^{n} r_i x^{i-1}$ the corresponding polynomial. With the syndromes $S_i = r(\alpha_i)\ \forall i = 1, \ldots, n-k$ (and the corresponding syndrome polynomial $S(x) = \sum_{i=1}^{n-k} S_i x^{i-1}$) we can determine the error–locator–polynomial (ELP)

$$\sigma(x) = \prod_{j \in \mathcal{J}} (x - \alpha_j), \tag{2}$$

where $\mathcal{J}$ is the set of error locations. The Key Equation (KE):

$$S(x) \cdot \sigma(x) \equiv \Omega(x) \mod x^{n-k}, \tag{3}$$

where $\deg \Omega(x) < n - k - \tau$ gives us the polynomial relation of the ELP and the syndrome polynomial. It is well–known that determining the ELP is equivalent to the problem of synthesizing the shortest linear feedback shift–register (LFSR) $\sigma(x)$ that generates the given syndrome sequence $\mathbf{S} = (S_1, S_2, \ldots, S_{n-k})$.

Definition 1 (Interleaved Reed–Solomon Code) *Let the set $\mathcal{K}$:*

$$\mathcal{K} = \{k_1, k_2, \ldots, k_s\},$$

978–1–4244–6017–5/10/$26.00 © 2010 IEEE

consist of s elements of $\mathbb{F}$, where all $k_i < n$.
An Interleaved Reed–Solomon code $\mathcal{IRS}(n, \mathcal{K}, s)$ of order s is given by

$$\mathcal{IRS}(n, \mathcal{K}, s) = \left\{ C = \begin{pmatrix} f_1(\mathcal{L}) \\ f_2(\mathcal{L}) \\ \vdots \\ f_s(\mathcal{L}) \end{pmatrix} : f_i \in \mathbb{F}_{k_i}[x] \right\}, \quad (4)$$

where $i = 1, \ldots, s$. If $k_i = k \ \forall i = 1, \ldots, s$ the IRS code is called *homogeneous*. The burst–error–correcting capability of a *heterogeneous* IRS code is given by

$$\tau_{IRS} = \frac{s}{s+1} \left(n - \frac{1}{s} \sum_{i=1}^{s} k_i \right). \quad (5)$$

B. Generalized Key Equation for Homogeneous IRS Codes Revisited

In this section we derive a "concatenated" key equation for homogeneous IRS codes in a polynomial way. It completes the approach of Feng and Tzeng [3, Section 1]. For a homogeneous IRS code $\mathcal{IRS}(n, \mathcal{K}, s)$, where we assume that burst–errors occur, we can formulate s key equations of the following form:

$$S^{(i)}(x)\sigma(x) + C^{(i)}(x)x^{n-k} = \Omega^{(i)}(x) \quad \forall i = 1, \ldots, s, \quad (6)$$

with the same ELP $\sigma(x)$.
Let us spread and shift to:

$$x^{i-1}S^{(i)}(x^s)\sigma(x^s) + x^{s(n-k)+i-1}C^{(i)}(x^s) = x^{i-1}\Omega^{(i)}(x^s), \quad (7)$$

for all $i = 1, \ldots, s$. Adding the s equations leads to:

$$\sigma(x^s) \underbrace{\left(\sum_{i=1}^{s} S^{(i)}(x^s)x^{i-1} \right)}_{\widetilde{S}(x)} \equiv \widetilde{\Omega}(x) \bmod x^{s(n-k)}, \quad (8)$$

where $\deg \widetilde{\Omega}(x) < s(n-k-\tau)$.

C. Basic Principle of FT's Euclidean Algorithm

First of all let's state the problem based on Equation (8) and then define a modified division algorithm.

Problem 1 (FT's Euclidean algorithm) *Given $(s+1)$ polynomials $\widetilde{S}(x)$ and $x^{s(n-k)+i} \ \forall i = 0, 1, \ldots, s-1$, then we search two polynomials $\sigma(x), \widetilde{\Omega}(x)$ such that* (8) *holds, where* $\deg \widetilde{\Omega}(x) < \deg \sigma(x^s)$.

Definition 2 (Congruence Class) *Let $a(x), b(x)$ be two polynomials over $\mathbb{F}_n[x]$, if $\deg a(x) \bmod s = \deg b(x) \bmod s$, then $a(x)$ is congruent to $b(x)$, denoted as $a(x) \sim b(x)$. A full congruence class consists of s congruence classes, i.e., $[x^i]$, for $i = 0, 1, \ldots, s-1$.*

The polynomials $\widetilde{S}(x)$ and $x^{s(n-k)+i}$ in (8) can be regarded as $r_j(x)$ and $b_j^{(i)}(x)$ when $j = 0$. $b_j^{(i)}(x), i = 0, 1, \ldots, s-1$ has a full congruence class of s. Thus one of them, say $b_j^{(v_j)}(x)$, where $v_j = \deg r_j \bmod s$, must be congruent to $r_j(x)$, namely, $b_j^{(v_j)}(x) \sim r_j(x)$. Additionally, if $\deg b_j^{(v_j)}(x) \geq \deg r_j(x)$, then the unique polynomials $q(x^s) \neq 0$ and $R(x)$ exit such that

$$b_j^{(v_j)}(x) = q(x^s)r_j(x) + R(x).$$

The division stops when $\deg R(x) < \deg r_j(x)$ if $R(x) \sim r_j(x)$ or when $R(x) \nsim r_j(x)$ for the first time. This is what we called modified division algorithm proceeded between two polynomials. By repeatedly applying the modified division algorithm, a generalized division algorithm [3, Section 2], is derived. It is based on the division of more than one polynomial.
Given $(s+1)$ polynomials $b_j^{(i)}(x)$ for $i = 0, 1, \ldots, s-1$ and $r_j(x)$, we consider the generalized division algorithm to obtain

$$b_j^{(v_j)}(x) = \tilde{p}_{j+1}(x^s)r_j(x) + \sum_{\substack{i=0 \\ i \neq v_j}}^{s-1} \tilde{q}_{j+1}^{(i)}(x^s)b_j^{(i)}(x) + r_{j+1}(x).$$

It can be rewritten as

$$r_{j+1}(x) = p_{j+1}(x^s)r_j(x) + \sum_{i=0}^{s-1} q_{j+1}^{(i)}(x^s)b_j^{(i)}(x), \quad (9)$$

where

$$\begin{aligned} p_{j+1}(x^s) &= -\tilde{p}_{j+1}(x^s), \\ q_{j+1}^{(i)}(x^s) &= -\tilde{q}_{j+1}^{(i)}(x^s), \quad \text{for } i \neq v_j, \\ q_{j+1}^{(v_j)}(x^s) &= 1. \end{aligned}$$

For $j \geq 1$, we have $b_{j+1}^{(v_j)}(x) = r_j(x)$, $b_{j+1}^{(i)}(x) = b_j^{(i)}(x), \forall i \neq v_j$ for updating since $b_j^{(v_j)}(x) \sim r_j(x)$ and $\deg b_j^{(v_j)}(x) > \deg r_j(x)$. At each update step,

$$r_j(x) = U_j(x^s)r_0(x) \mod x^{s(n-k)} \quad (10)$$

fulfills, where

$$U_{j+1}(x) = p_{j+1}(x)U_j(x) + \sum_{i=0}^{s-1} q_{j+1}^{(i)}(x)V_j^{(i)}(x).$$

The update rule is $V_{j+1}^{(v_j)}(x) = U_j(x)$, $V_{j+1}^{(i)}(x) = V_j^{(i)}(x), \forall i \neq v_j$ [3, Section 3]. $U_0(x)$ and $V_0^{(i)}(x)$ are initialized as 1 and 0 respectively. If $\deg r_{j-1}(x) \geq \deg U_{j-1}(x^s)$ and $\deg r_j(x) < \deg U_j(x^s)$, let $k = j$, $U_k(x) = \sigma(x)$.
An example is considered in the appendix.
Note that for $s = 1$, FT's Euclidean algorithm becomes the classic Euclidean algorithm and the modified division based on the congruence classes simplifies to the normal division with only two polynomials.

III. Virtual Extension to an IRS code

A. Basic Principle

We shortly describe the Schmidt–Sidorenko–Bossert scheme [11]. The basic idea is the virtual extension of a RS code to an Interleaved Reed–Solomon (IRS) code. This IRS code is denoted by $\mathcal{VIRS}(n, k, s)$, where n and k are the

parameters of the original $\mathcal{RS}(n,k)$ code. The parameter s denotes the order of interleaving of the heterogeneous IRS code.
Let $p(x) = \sum_{j=0}^{n-1} p_j x^j$ be a univariate polynomial in $\mathbb{F}_n[x]$. Then $p^{<i>}(x) \in \mathbb{F}_n[x]$ represents the polynomial

$$p^{<i>}(x) = \sum_{j=0}^{n-1} p_j^i x^j,$$

in which each coefficient is raised to the power i. The virtually extended IRS code is defined as follows.

Definition 3 (Virtual Extension to an IRS code) *Let $\mathcal{RS}(n,k)$ be a Reed–Solomon code with the evaluation polynomial $f(x)$ as defined in (1). The virtually extended Interleaved Reed–Solomon code $\mathcal{VIRS}(n,k,s)$ of order s is given by $\mathcal{VIRS}(n,k,s) =$*

$$\begin{pmatrix} \mathbf{c}^{<1>} \\ \mathbf{c}^{<2>} \\ \vdots \\ \mathbf{c}^{<s>} \end{pmatrix} = \begin{pmatrix} f(\mathcal{L}) & : f(x) \in \mathbb{F}_k[x] \\ f^2(\mathcal{L}) & : f^2(x) \in \mathbb{F}_{2(k-1)+1}[x] \\ \vdots & \\ f^s(\mathcal{L}) & : f^s(x) \in \mathbb{F}_{s(k-1)+1}[x] \end{pmatrix}. \quad (11)$$

Clearly, the parameter s must satisfy $s(k-1)+1 \leq n$. The scheme is restricted to low–rate Reed–Solomon codes and allows to decode beyond half the minimum distance. The virtual extension of order 3 is illustrated for an $\mathcal{RS}(31,4)$ code in Figure 1, where the information length of the i-th

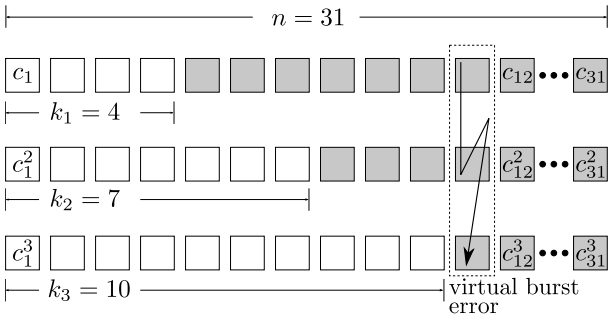

Fig. 1. Illustration of the virtual extension $\mathcal{VIRS}(31,4,3)$ of an $\mathcal{RS}(31,4)$ code with interleaving factor $s = 3$.

codeword is $k_i = i(k-1)+1$. The decoding procedure for the virtual extension of an RS code is as follows: the received word $\mathbf{r} = \mathbf{c} + \mathbf{e}$ is raised element per element $(\mathbf{r}^{<2>}, \mathbf{r}^{<3>}, \ldots, \mathbf{r}^{<s>})$ and the mentioned heterogeneous IRS code is obtained. Clearly, through the virtual extension, the error is also "extended" and every single received word $\mathbf{r}^{<i>}$ is at the same position erroneous (*virtual burst error*). Due to the additional equations, the decoding radius is increased to:

$$\tau = \left\lfloor \frac{sn - \binom{s+1}{2}(k-1) - s}{s+1} \right\rfloor. \quad (12)$$

The radius τ is greater than $\lfloor (n-k)/2 \rfloor$ for Reed–Solomon codes with code rate $R < 1/3$. (For further details of this scheme, see [11]). We remark that the rate–restriction and the increased decoding radius are similar to the original Sudan algorithm [9], [10]. Figure 2 illustrates the decoding procedure

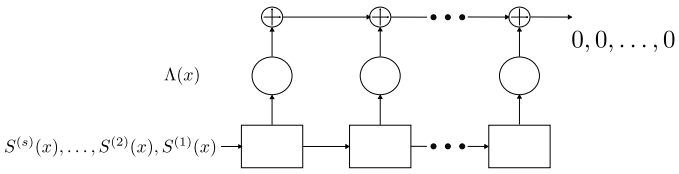

Fig. 2. The multi–sequence shift–register problem.

as a multi–sequence shift–register problem. If $s = 1$ the BM or the SKHN algorithm can be used to solve the single–sequence problem.

IV. Sequence Shifting for the Virtual Extension

In this section, we investigate the decoding problem for the virtual extension to an IRS code. The multi–sequence problem of varying length is reformulated and the reformulation is proved.

Problem 2 (Multi–sequence equal length) *Let s sequences $\mathbf{S}^{(h)} = (S_0^{(h)}, S_1^{(h)}, \ldots, S_{N-1}^{(h)}), h = 0, \ldots, s-1$ of the same length N be defined over $\mathbb{F}$. Then we search the connection polynomial $\sigma(x) = \sigma_0 + \sigma_1 x + \cdots + \sigma_{\ell-1} x^{\ell-1} + x^\ell$ with the smallest degree ℓ such that:*

$$S_i^{(h)} + \sigma_{\ell-1} \cdot S_{i-1}^{(h)} + \cdots + \sigma_0 \cdot S_{i-\ell}^{(h)} = 0 \quad (13)$$

for all $i = \ell, \ell+1, \ldots, N-1$ and for all $h = 0, \ldots, s-1$ holds.

Problem 3 (Multi–sequence varying length) *Let s sequences $\mathbf{S}^{(h)} = (S_0^{(h)}, S_1^{(h)}, \ldots, S_{N_h-1}^{(h)})$ of different lengths $N_0, N_1, \ldots, N_{s-1}$ be defined over $\mathbb{F}$. Then we search the connection polynomial $\sigma(x) = \sigma_0 + \sigma_1 x + \cdots + \sigma_{\ell-1} x^{\ell-1} + x^\ell$ with smallest degree ℓ such that:*

$$S_i^{(h)} + \sigma_{\ell-1} \cdot S_{i-1}^{(h)} + \cdots + \sigma_0 \cdot S_{i-\ell}^{(h)} = 0 \quad (14)$$

for all $i = \ell, \ell+1, \ldots, N_h - 1$ and for all $h = 0, \ldots, s-1$ holds.

In the following we assume that that the minimal length ℓ is smaller than or equal to $N_{min} = \min_i N_i$.
From the s sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(s-1)}$ of Problem 3 with different lengths we define

$$\widetilde{s} = s + \sum_{i=0}^{s-1} (N_i - N_{min}) \quad (15)$$

sequences $\widetilde{\mathbf{S}}^{(h,j)}$ with the same length N_{min} in the following manner:

$$\begin{aligned} \widetilde{\mathbf{S}}^{(h,j)} &= (S_j^{(h)}, S_{j+1}^{(h)}, \ldots, S_{j+N_{min}-1}^{(h)}) \\ &= (\widetilde{S}_0^{(h,j)}, \widetilde{S}_1^{(h,j)}, \ldots, \widetilde{S}_{N_{min}-1}^{(h,j)}) \end{aligned} \quad (16)$$

for all $h = 0, \ldots, s-1$ and $j = 0, \ldots, N_h - N_{min}$.

Proposition 1 (Reformulated multi–sequence problem) *In the following we assume that the degree ℓ of the connection polynomial $\sigma(x)$ is smaller than or equal to N_{min}. Given the $\widetilde{s}$ sequences $\widetilde{\mathbf{S}}^{(0,0)}, \ldots, \widetilde{\mathbf{S}}^{(s-1,N_{s-1}-N_{min})}$ with equal length N_{min} as defined in Equation (16). Then the connection polynomial $\sigma(x)$ for these sequences solving Problem 2 is always the same as the one solving Problem 3 for the original s sequences $\mathbf{S}^{(0)}, \ldots, \mathbf{S}^{(s-1)}$.*

Proof: Clearly, we have the following relation from Definition (16):

$$\widetilde{S}_i^{(h,j)} = S_{i+j}^{(h)}, \tag{17}$$

for all $i + j = 0, 1, \ldots, N_h - 1$.
For the sequences $\widetilde{\mathbf{S}}^{(h,0)}, \widetilde{\mathbf{S}}^{(h,1)}, \ldots, \widetilde{\mathbf{S}}^{(h,N_h-N_{min})}$ and for a connection polynomial $\sigma(x) = \sigma_0 + \sigma_1 x + \cdots + \sigma_\ell x^\ell$ ($\sigma_\ell = 1$) we have the following relation after solving Problem 2:

$$\widetilde{S}_i^{(h,j)} + \sigma_{\ell-1} \cdot \widetilde{S}_{i-1}^{(h,j)} + \cdots + \sigma_0 \cdot \widetilde{S}_{i-\ell}^{(h,j)} = 0, \tag{18}$$

for all $i = \ell, \ldots, N_{min} - 1$ and $j = 0, \ldots, N_h - N_{min}$ and $h = 0, \ldots, s-1$. With Equation (17) we can write:

$$S_\iota^{(h)} + \sigma_{\ell-1} \cdot S_{\iota-1}^{(h)} + \cdots + \sigma_0 \cdot S_{\iota-\ell}^{(h)} = 0, \tag{19}$$

for all $\iota = \ell, \ldots, N_h - 1$ and $h = 0, \ldots, s-1$. That is the original multi–sequence shift–register problem of varying length. ∎

Clearly, the complexity of solving the equivalent multi–sequence problem of equal length will be higher. We investigate it for the special case of the virtual extension to an IRS code in the following section.

V. Complexity Analysis for the Virtual Extension

Feng and Tzeng did not analyze the time complexity in their contribution [3]. We assume a time complexity for s sequences with the same length N of $\mathcal{O}\left(sN^2\right)$.
For the virtual extension to an IRS code of Section III with interleaving factor s, we obtain s sequences with length $N_i = n - k - (i-1)(k-1)\ \forall i = 1, \ldots, s$.
With the reformulation of the previous section, the length of all new sequences equals the length of the shortest one and is

$$N_{min} = N_s = n - k - (s-1)(k-1). \tag{20}$$

Let us calculate the number of new sequences $\widetilde{s}$ of equal length N_{min} more explicitly:

$$\begin{aligned}\widetilde{s} &= s + \sum_{i=1}^{s} (N_i - N_s) \\ &= s + \sum_{i=1}^{s-1} (s-1)(k-1) - (i-1)(k-1) \\ &= s + \binom{s}{2}(k-1). \end{aligned} \tag{21}$$

The overall complexity of the IRS–scheme is $\mathcal{O}\left(\widetilde{s} N_{min}^2\right)$, that is:

$$\begin{aligned}&= \mathcal{O}\left((s + \binom{s}{2}(k-1)) \cdot (n - k - (s-1)(k-1))^2 \right) \\ &\approx \mathcal{O}\left(s^2 k \cdot (n-k)^2\right). \end{aligned} \tag{22}$$

VI. Generalized Sequence Shifting

Let us again consider the general multi–sequence shift–register problem of varying length. In Section IV we assumed that the degree of connection polynomial $\sigma(x)$ (and so the length of the generating shift–register) is guaranteed to be smaller than or equal to the shortest sequence of Problem 3. This holds for the virtually extended RS code. Nevertheless, the more general case needs a small modification.
Algorithm 1 is our proposed method. It is well–known that for a given shift–register $\sigma(x)$ with degree ℓ solving Problem 3 any sequence of length smaller than or equal to ℓ can be added to the problem set and will be generated by the given $\sigma(x)$. So if the degree of the connection polynomial $\sigma(x)$ is equal to the length of the actual smallest sequence, returned by Algorithm 2, then we restart the generation without taking into account the shortest sequence.
Note, any algorithm solving a multi–sequence shift–register problem can be used as `ShiftEqual`.

Algorithm 1: Multi–Sequence Shift–Register Analysis of Varying Length

Input: Sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m-1)}$ of length $N_0 \geq N_1 \geq \cdots \geq N_{m-1}$
Output: Shortest Shift–Register $\sigma(x)$ of degree ℓ generating $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m-1)}$

Initialize:
Arbitrary Shift–Register $\sigma(x)$ of degree N_{m-1};
Integers $\binom{N}{\kappa} \leftarrow \binom{N_{m-1}}{0}$;

```
while (N == N_{m-1-κ}) do
    σ(x) ← Shift(S^(0), S^(1), ..., S^(m-1-κ));
    N ← deg σ(x);
    κ ← κ + 1;
```

Algorithm 2: `Shift`$(\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m-1)})$

Input: Sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(m-1)}$ of length $N_0 \geq N_1 \geq \cdots \geq N_{m-1}$
Output: Intermediate Shift–Register $\sigma(x)$

Initialize: Calculate $\widetilde{m}$ according to Equation (15)

```
for i = 0 to m̃ − 1 do
    Initialize S̃^(i) according to Equation (16);
σ(x) ← ShiftEqual(S̃^(0), S̃^(1), ..., S̃^(m̃−1));
```

VII. Conclusion

We derived the generalized key equation of Feng–Tzeng for the multi–sequence shift–register analysis in an alternative polynomial way. For our new decoding approach up to the Sudan radius, we used the combination of a virtually

TABLE I
FT'S EUCLIDEAN ALGORITHM FOR A $\mathcal{VIRS}(10,3,2)$ WITH SHIFTED SEQUENCES.

j	$r_j(x)$	$p_j(x)$	$q_j^{(0)}(x)$	$q_j^{(1)}(x)$	$q_j^{(2)}(x)$	$q_j^{(3)}(x)$	$U_j(x)$
0	$2x^{19}+9x^{18}+5x^{17}+8x^{16}+9x^{15}+5x^{14}+3x^{12}+5x^{11}+8x^9+8x^6+x^5+3x^4+8x^3+x^2+4x+3$	–	–	–	–	–	1
1	$4x^{18}+3x^{17}+4x^{15}+3x^{14}+7x^{13}+4x^{12}+3x^{11}+7x^{10}+x^9+4x^8+7x^7+x^6+3x^5+8x^4+7x^3+5x^2+9x+4$	$5x+5$	4	8	10	1	$5x+5$
2	$3x^{16}+9x^{13}+5x^{12}+9x^{10}+3x^9+8x^8+2x^7+8x^6+3x^5+4x^4+5x^3+2x^2+8x+6$	$8x+8$	0	9	1	6	$7x^2+3x+2$
3	$8x^{17}+2x^{16}+8x^{14}+10x^{13}+x^{12}+3x^{11}+x^{10}+10x^9+6x^8+2x^7+3x^6+x^5+9x^4$	$7x$	1	0	0	0	$5x^3+10x^2+3x$
4	$2x^{15}+8x^{14}+8x^{13}+x^{12}+9x^{10}+9x^9+3x^8+6x^7+8x^6+7x^5+6x^4+10x^3+4x^2+5x+1$	$4x+1$	x	1	3	0	$9x^4+8x^3+3x^2+9x+4$

extended RS code and a reformulation of the corresponding multi–sequence shift–register problem of varying length for the heterogeneous IRS code. The reformulated problem was solved by the generalized Euclidean algorithm of Feng and Tzeng.

The reformulation to a multi–sequences problem of equal length was proved. The increased complexity was analyzed.

REFERENCES

[1] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes," *Information and Control*, vol. 27, no. 1, pp. 87–99, 1975.

[2] E. R. Berlekamp, *Algebraic coding theory*. McGraw-Hill, 1968.

[3] G. L. Feng and K. K. Tzeng, "A Generalized Euclidean Algorithm for Multisequence Shift-Register Synthesis," *Information Theory, IEEE Transactions on*, vol. 35, no. 3, pp. 584–594, 1989. [Online]. Available: http://dx.doi.org/10.1109/18.30981

[4] ——, "A Generalization of the Berlekamp-Massey Algorithm for Multisequence Shift-Register Synthesis with Applications to Decoding Cyclic Codes," *Information Theory, IEEE Transactions on*, vol. 37, no. 5, pp. 1274–1287, 1991. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=133246

[5] G. Schmidt and V. R. Sidorenko, "Linear Shift-Register Synthesis for Multiple Sequences of Varying Length," May 2006. [Online]. Available: http://arxiv.org/abs/cs/0605044

[6] L. Wang, "Euclidean Modules and Multisequence Synthesis," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, ser. Lecture Notes in Computer Science, S. Boztaş and I. E. Shparlinski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, October 2001, vol. 2227, ch. 25, pp. 239–248. [Online]. Available: http://dx.doi.org/10.1007/3-540-45624-4_25

[7] V. Y. Krachkovsky, "Reed-Solomon Codes for Correcting Phased Error Bursts," *Information Theory, IEEE Transactions on*, vol. 49, no. 11, pp. 2975–2984, 2003. [Online]. Available: http://dx.doi.org/10.1109/TIT.2003.819333

[8] D. Bleichenbacher, A. Kiayias, and M. Yung, "Decoding Interleaved Reed-Solomon Codes over Noisy Channels," *Theor. Comput. Sci.*, vol. 379, no. 3, pp. 348–360, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2007.02.043

[9] M. Sudan, "Decoding of Reed-Solomon Codes beyond the Error-Correction Bound," *Journal of Complexity*, vol. 13, no. 1, pp. 180–193, March 1997. [Online]. Available: http://dx.doi.org/10.1006/jcom.1997.0439

[10] A. Zeh, S. Kampf, and M. Bossert, "On the Equivalence of Sudan-Decoding and Decoding via Virtual Extension to an Interleaved Reed-Solomon Code," in *8th International ITG Conference on Source and Channel Coding*, Siegen, Germany.

[11] G. Schmidt, V. Sidorenko, and M. Bossert, "Decoding Reed-Solomon Codes Beyond Half the Minimum Distance using Shift-Register Synthesis," in *Information Theory, 2006 IEEE International Symposium on*, 2006, pp. 459–463. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4036003

APPENDIX

We consider a $\mathcal{RS}(10,3)$ code over $\mathbb{F} = GF(11)$. For the virtual extension of Section III with interleaving factor $s = 2$ the decoding radius can be increased to $\tau = 4$ according to Equation (12).

One can take $\alpha_i = 2^{i-1} \mod 11$, $i = 1, 2, \ldots, 10$ with $k = 3$ and $f(x) = x$ we obtain the corresponding codeword:

$$\mathbf{c} = (1, 2, 4, 8, 5, 10, 9, 7, 3, 6),$$

We add an error-word of weight $\tau = 4$ and obtain the following received word $\mathbf{r}^{<1>}$ and its squared counterpart $\mathbf{r}^{<2>}$:

$$\mathbf{r}^{<1>} = (10, 9, 9, 7, 5, 10, 9, 7, 3, 6)$$
$$\mathbf{r}^{<2>} = (1, 4, 4, 5, 3, 1, 4, 5, 9, 3).$$

By evaluating $\mathbf{r}^{<1>}$ and $\mathbf{r}^{<2>}$, we obtain the two syndromes:

$$\mathbf{S}^{(1)} = (2, 9, 5, 0, 8, 1, 4),$$
$$\mathbf{S}^{(2)} = (8, 3, 0, 3, 3).$$

Obviously, $\mathbf{S}^{(2)}$ has the shortest length $N_{min} = n - k - (s-1)(k-1) = 5$. From the sequence $\mathbf{S}^{(1)}$ we construct three sequences of length N_{min} by shifting, as follows:

$$\widetilde{\mathbf{S}}^{(0,0)} = (2, 9, 5, 0, 8),$$
$$\widetilde{\mathbf{S}}^{(0,1)} = (9, 5, 0, 8, 1),$$
$$\widetilde{\mathbf{S}}^{(0,2)} = (5, 0, 8, 1, 4),$$
$$\widetilde{\mathbf{S}}^{(1,0)} = (8, 3, 0, 3, 3).$$

These reconstructed syndromes can be applied as the input to FT's Euclidean algorithm. Here, the intermediate results according to this algorithm are listed in Table I.

Since $\deg U_4(x^4) = 16 > \deg r_4(x) = 15$, we have $k = 4$ and the monic polynomial $\delta U_4(x) = x^4 + 7x^3 + 4x^2 + x + 9$ is the ELP. It can be easily checked that $2^0, 2^1, 2^2$, and 2^3 are roots of $\delta U_4(x)$. Hence the received word is with errors in the first, second, third and fourth positions.